Übungsaufgaben zur Vertiefung des Themas Rechnerarchitekturen im Kurs 5CS-OPSY-30

Arbeitsmaterial:

* Skript zur LV
* Internetrecherchen

Aufgaben:

1 Welche konzeptionelle Rechnerarchitektur liegt auch heutigen PCs und Servern meist zugrunde?

Von-Neumann-Architektur

2 Welche Bestandteile umfasst diese Architektur, welche Vorteile besitzt sie und welchen entscheidenden Nachteil?

Rechenwerk ALU, Steuerwerk Control Unit, Speicherwerk Memory Unit, Ein/Ausgabewerk I/O-Unit, Bus-System

Sequenzielle Bearbeitung vermeidet Inkonsistenzen, gemeinsamer Speicher für Daten und Befehle, Befehle können im laufenden Betrieb geändert werden, Programmschleifen möglich, Programmierung und Modell gut beherrschbar, Turing-mächtig

Nachteil Von-Neumann-Flaschenhals, Bussystem leicht überlastet und zu langsam

3 Durch welche technologischen Maßnahmen wird diese Rechnerarchitektur in modernen Computern ergänzt, die konzeptionell eigentlich nicht zu ihr gehören?

Parallele Bearbeitung, entweder mit Multiprozessoren oder in multiplen Ausführungseinheiten (RISC-Prozessoren), Trennung von Daten und Befehlen durch jeweils eigene Cache-Speicher

4 Erläutern Sie den Nutzen, den das Einfügen von Zwischenspeicher (Cache) zwischen die CPU und den Arbeitsspeicher bringt

Cache speichert wenige aber häufig benutzte Daten und Befehle in schnellem SRAM auf der CPU zwischen. Daten und befehle können mit hoher Geschwindigkeit ausgelesen werden, wenn sie im Cache vorhanden sind. Beschleunigt Prozesse erheblich

5 Wie unterscheiden sich RISC- und CISC-Prozessoren? Welchen Nutzen bringt die RISC-Architektur?

Konventionelle Prozessoren (nachträglich CISC genannt) versuchen durch komplexe Befehlssätze auch komplexe Aufgaben durch einen einzigen Maschinenbefehl abrufen zu lassen. Vereinfacht die Programmierung. Problem: Für die Interpretation des Befehls und die anschließende Ausführung werden viele Taktzyklen benötigt, Leistungsengpass. RISC-Prozessoren haben nur einen kleineren Satz von Befehlen, die aber nicht interpretiert werden müssen und in einzelnen Taktzyklen abgearbeitet werden. Programmieraufwand und Speicherbedarf höher, aber schnellere Abarbeitung und daher höhere Gesamtleistung

6 Warum werden nichtflüchtige Massenspeicher (Festplatten und SSD) nicht Bit- oder Byteweise gelesen und geschrieben, sondern in Sektoren bzw. Blöcken? Warum vereinfacht dies die Herstellung von SSD-Laufwerken?

Festplatten haben die Zugriffsweise von Bändern übernommen, es musste vor- und zurückgespult werden. Um nicht jedes Byte einzeln adressieren zu müssen und damit Platz zu verschwenden, wurden Bytes zu Blöcken zusammengefasst und je Block eine Adresse vergeben Nachteil: es muss immer der ganze Block gelesen und geschrieben werden. Nutzen bei SSD: NAND Flash mit spaltenweisem Auslesen reicht, weil nur ganze Spalten als Block verarbeitet werden müssen. NOR-Fash ist nicht erforderlich.

7 Warum überstehen SSD-Laufwerke usw. mit MLC-Speicher deutlich weniger Schreibvorgänge als solche mit SLC-Speicher?

Im SLC-Speicher sind nur zwei Zustände je Zelle definiert, Ein oder aus. Der Übergang liegt irgendwo im Bereich von 50% der maximalen Ladungsmenge im Floating Gate. Wenn mit der Nutzungsdauer die Ladungsmenge auf 80% oder weniger abnimmt, ist dies immer klar noch als ein oder aus erkennbar. Bei MLCs gibt es mehrere Level, z.B. 4. Wird die Ladungsmenge nach längerer Nutzung geringer, ist irgendwann keine Unterscheidung der höheren Level mehr möglich, die Zelle wird unbrauchbar.

8 Mit welchen Angaben können Sie die voraussichtliche Lebensdauer von SSD-Laufwerken miteinander vergleichen?

Total Bytes Written oder Terabytes Written, gibt die Anzahl der Schreibzyklen in Terabytes an, die ein SSD-Laufwerk mindestens übersteht.